Department of Physics and Astronomy, University of Canterbury,
Private Bag 4800, Christchurch, New Zealand

Real Data tests of the IceCube Verification Software incl. Installation Guide for IceVerV00-02-06

Matthias Danninger

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

GCAS Personal Research Project 06/07

Supervisor: Dr. Jenni Adams

Co-supervisor: Dr. Surujhdeo Seunarine

Abstract

This project explains how to set up a working platform for the IceCube software IceVer. As a functional check of the software, ice properties of the Ice sheet at the South-pole are researched with the interstring flasher verification project. Relative string positions of the 2006 IceCube configuration are determined out of flasher-data and compared to the real grid positions.

Contents

1 Introduction **1**

1.1 IceCube 1

1.2 Optical Modules and LED Flashing 2

2 Running the Flasher-Verification Project **4**

2.1 Installations 5

2.2 Running the Flasher Verification Macro 6

3 Results **7**

3.1 IceTray Event Viewer 7

3.2 Real Data flasher-runs analyzed with the flasher-verif macro 7

4 Interpretation **13**

4.1 Relative String Positions of IceCube 2006 Configuration 13

4.2 DOM Sensitivity 15

5 Conclusion **16**

Bibliography **17**

Acknowledgments **17**

List of Figures

1.1 IceCube Digital Optical Module 2

3.1 Reconstructed Muon Event in the current Configuration of IceCube 8
3.2 Data Set 1 10
3.3 Data Set 2 11
3.4 Data Set 3 12

4.1 Grid-plan of the complete planned IceCube site 13
4.2 Reconstructed grid 14

Chapter 1

Introduction

1.1 IceCube

The neutrino telescope, IceCube, is designed for the detection of high and ultra-high-energy cosmic neutrinos. The researched neutrinos are elementary particles with no electrical charge and barely any mass. The aim of IceCube is to detect neutrinos generated by extra terrestrial sources. Neutrinos, produced in the decays of highly accelerated particles in intergalactic magnetic fields, are a researched source. As well as WIMP annihilation is also predicted as a source of neutrinos, the detector is used to research particles beyond the standard model of particle physics. Neutrinos, ν, open up a new window to the cosmos, as they propagate undisturbed through the universe, therefore allowing an undistorted view in contrast to messengers like photons or protons [1]. The detector is currently under construction and is expected to be completed in early 2011. It will consist of 80 strings, each instrumented with 60 Digital Optical Modules (DOM), to be capable of detecting signals over a wide dynamic range. This region ranges from single photons to several thousand photons arriving within microseconds of each other. The 4800 DOMs are installed along the strings and deployed in the West Antarctic Ice Sheet at a depth between 1450 m and 2500 m. This in-ice array is deployed in a triangular grid pattern with a characteristic spacing of 125 m, enclosing an area of 1 km^2. Therefore the completed IceCube detector will instrument a volume of 1 km^3.

The very small neutrino-particle cross section allows neutrinos to pass the Earth virtually unhindered, and they can be therefore detected as “up coming” neutrinos in IceCube. Muons, μ, are produced in charged current interactions with nuclei, like protons, as ice is the dominant medium inside, or close to the detector. The DOMs detect the Cherenkov light radiated by passing high energetic μ. The Cherenkov effect occurs only when a charged particle travels through a medium with a speed greater than the speed of light in that particular medium. The front of the light-cone that is created has a characteristic angle, and is shaped like a Mach-cone. The specially shaped light cone contains all information required to calculate the muon track from the recorded DOM signals. The reconstructed muon track provides the information to calculate the progenitor neutrino.

1.2 Optical Modules and LED Flashing

Each string is connected to a junction box, located at the surface above the detector. From the surface junction box a cable carries all string cables to the counting house. The signals are already digitized and time stamped in the DOMs [2]. A key design feature of the detector are the DOMs, which operate as a complete and autonomous data acquisition system. The fundamental detector element of IceCube is a 25 cm diameter Photo Multiplier Tube (PMT) assembled with a suite of electronics within a special glass pressure housing of 35.6 cm diameter, shown in Figure 1.1.

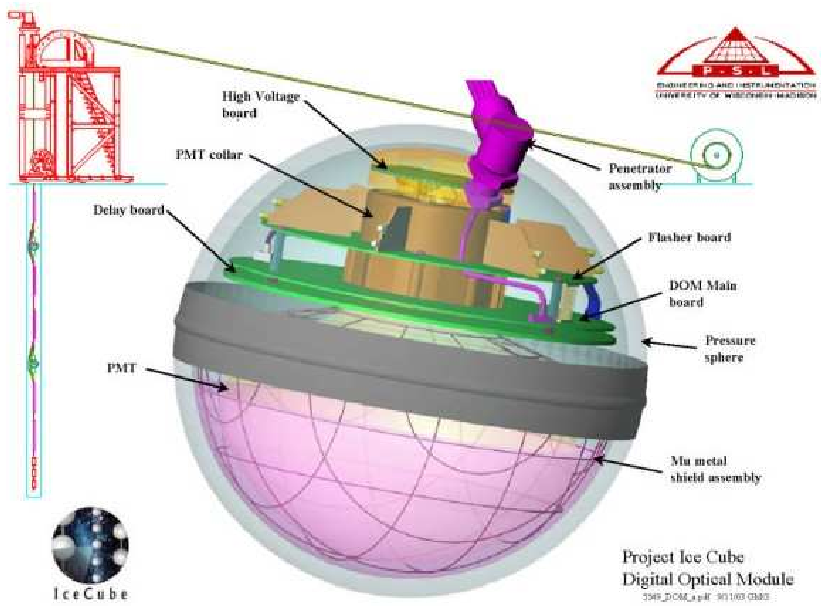

Figure 1.1: *A schematic view of an IceCube DOM [2]*

After being deployed in the ice, DOMs cannot be changed or repaired, as inaccessibility and the extreme high-pressure environment constrain handling of the DOM components. Only, highly reliable, power efficient and intensively tested components are integrated in the DOMs. All modules are remotely controlled independently of each other and synchronized by a master clock system. Additionally, calibration tools are installed on board. The LED flasher board contains 12 gallium nitride LEDs pointing radially outward from the DOM, 6 of which are positioned horizontally, and the other 6 pointing up at an angle of 48 degree [2]. The peak operating wave-length ranges from 400 nm - 420 nm. The optical beacon can repeat the signals at a rate of 610 Hz. This artificial light source is used for following calibration and verification modes[3].

(1) calibration of local coincidence, timing and geometry
(2) interstring timing and geometry calibration
(3) verification of optical properties of the ice
(4) linearity calibration of surrounding DOMs
(5) high energy cascade calibration

The actual Cherenkov photons are detected with the PMT, creating analog charge signals, which are digitized by a Fast Analog-to-Digital Converter (FADC) and a set of three Advanced Transient Waveform Digitizer (ATWD). The ATWD contains three channels with gains of $\frac{1}{4}$, 2 and 16 [1]. The ATWDs are sampling 128 bins of 3.3 ns width, whereas the FADC samples at a rate of 40 MHz for $6.4\mu s$. This raw IceCube data contains time stamped and digitized waveforms of the measured charge pulses.

Chapter 2

Running the Flasher-Verification Project

The IceTray software, based on the C++ language, is programed as a Frame/Stream/Stop model to provide a conceptual framework in which data can be analyzed [4]. The concept of a Frame is similar to a frame in a video. This Frame is an electronic picture of a certain time interval, configuration and geometry of IceCube. The electronic picture created is analogous to a single frame in a continuous electronic movie of the experiment. These frames contain many different pieces of data, such as the detector geometry, timing and voltage. Records are created, if one of these components is changing at the same time. Grouping all records of the same type defines a stream. It is called a stop, when a stream is combined with time-like information, because this is usually the point where the experiment is stopped and the information is provided to the user (i.e. a muon event in the detector).
The executable files are written as python scripts, also called a macro, consisting of many standard building blocks. The standard building blocks of the IceCube offline analysis software are the I3-Modules. The streams defined in IceTray are named Geometry, Calibration, Detector Status and Physics. An I3-Module tells IceTray which streams it is interested in by implementing the virtual functions Geometry(I3FramePtr), Calibration(I3FramePtr), DetectorStatus(I3FramePtr), or Physics(I3FramePtr). These I3-Modules have to be ordered in a certain way, because some Modules need the output variable of another Module as their input variable. This process is also called serialization of the I3-Modules, which can be viewed with the dataio-shovel '.i3' file browser of each macro.
dataio-shovel /data/test-data/filename.i3

The actual installation process involves several steps, which are explained in detail for the example of the flasher-verif macro in the next section [5]:
1. Installing various prerequisites on your machine and verifying that they work
2. Checking out code
3. Building your own set of tools that the code requires to build
4. Running a few commands to generate an environment script, and
5. Building the software.

2.1 Installations

In order to achieve the task of installing and running IceVer, the IceCube verification software, this project started with a blank computer. The first step is to install a supported operating system, like Linux Redhat Enterprise 3 or 4. Secondly, root was downloaded from the Cern web page:

(http://root.cern.ch/)

Currently ROOT v5.12.00.is provided in many precompiled binary versions. Downloading and installing the precompiled binary versions requires the exact named version of the platform (i.e. in this case gcc ver. 3.4.4-2), otherwise the source file has to be downloaded and personally compiled. Thirdly, an installation of Java is required to build some offline software. The only requirement is to have the Java SDK v1.4.2 from sun **(java.sun.com)** installed. Additionally, the JAVA HOME environment variable should be set to:

export JAVA_HOME=/usr/java/j2sdk1.4.2

The next step is to checkout and build the i3-tools. It is therefore necessary to have several gigabytes of disc space available. The following command lines build the tool management system, and download, compile and install all i3-tools.

svn co SVN/tools/DarwinPorts/trunk_ports_src
cd ports_src
./i3-install.sh I3_PORTS

As a last step in setting up a working environment for the IceCube software, the IceVer meta-project has to be downloaded from the Wisconsin server with the command line:

svn co http://code.icecube.wisc.edu/svn/meta-projects/IceVer/releases/V00-02-06/ IceVerV00-02-06

Finally the working environment has to be set up and compiled with:

cd IceVer-V00-02-06
make env.sh
source env.sh
make

2.2 Running the Flasher Verification Macro

Before initiating the IceCube verification software (IceVer), one has to set up the working environment with the following command:

cd IceVer-V00-02-06
source env.sh

The flasher-verif scripts directory contains 4 different macros. The two python scripts, FatFlasherVerif.py and FlasherVerif.py, take as data input the same files, and create a ROOT file in the output directory, which is the current working directory. When the macros are run, as described in the command line below, the created ROOT output file contains histograms and ntuples. They can either be viewed in ROOT or using the two macros, MakePlots.C and RelativeDepth.C, which are also listed in the scripts directory. These can be run to create an output.png file, containing a table of all relevant diagrams.
The macro is run in the following way:

flasher-verif/resources/scripts/FatFlasherVerif.py [indir] [runtag]
python /IceVerV00-02-06/flasher-verif/resources/scripts/FatFlasherVerif.py /icecube/soft data/cv-flashers/ SPS-TESTDAQ01_run0027567_Flasherboard-LC-38-04-Megalophobia-PLUS-30-16-Acura_Integra-PLUS-38-28-Zoophobia-PLUS-30-40-Solna-FBOUT-VER-Bright127-Wid th100-Mask001-Rate0009-Deadtime051200-ATWD0 outputfile.root

It must be ensured that the path for the test-data file or later on the real-data file is chosen correctly. Depending on the python script run, either a **flasher-timing-runnum.root** or **outputfile.root** file are created. They can be previewed in ROOT with the following commands:

root -l outputfile.root
[root] TBrowser tb

or the ROOT script MakePlots.C can be chosen to run with the following command, to create the **outputfile.png** file.

root 'MakePlots.C("filename", "outputname")'
root ' /IceVerV00-02-06/flasher-verif/resources/scripts/MakePlots.C("outputfile.root","pr

Chapter 3

Results

3.1 IceTray Event Viewer

The IceTray event viewer allows one to demonstrate the actual reconstructed event from the DOM data. Only files labeled with the ending '.i3' can be viewed with the following command line:

eventviewer-standalone /filename.i3

Figure 3.1. is created with the event viewer and shows a reconstructed muon event from above the detector. The black dotted lines represent the currently deployed strings, whereas IceTop is illustrated by the short black dashes above the detector strings. The green diagonal line describes the calculated muon event. The colors of the responding DOMs represent the time sequence of the event. The DOMs activated the earliest, are displayed with reddish colors. The time-like evolution of the event is represented in the order yellow, green to blue, which is the sequence generally describing the increasing energy of visible light.

3.2 Real Data flasher-runs analyzed with the flasher-verif macro

Figures 3.2, 3.3 and 3.4 show the output from the flasher-verif software for three different flasher runs of IceCube in 2006. All plots are obtained from the generated **output.root** file. The root 'Tree-viewer' is used to create single related plots out of the automatically generated plots. In order to determine additional information, such as the relative distances between strings and distances between flasher and reviver strings, the following command structure was used:

[root] TBrowser tb

Open root outputfile with the file browser and right click with the mouse on the displayed flasher folder, to open it with the TreeViewer 'StartViewer'. After going back to the root terminal, the following command samples can be executed to view the plots:

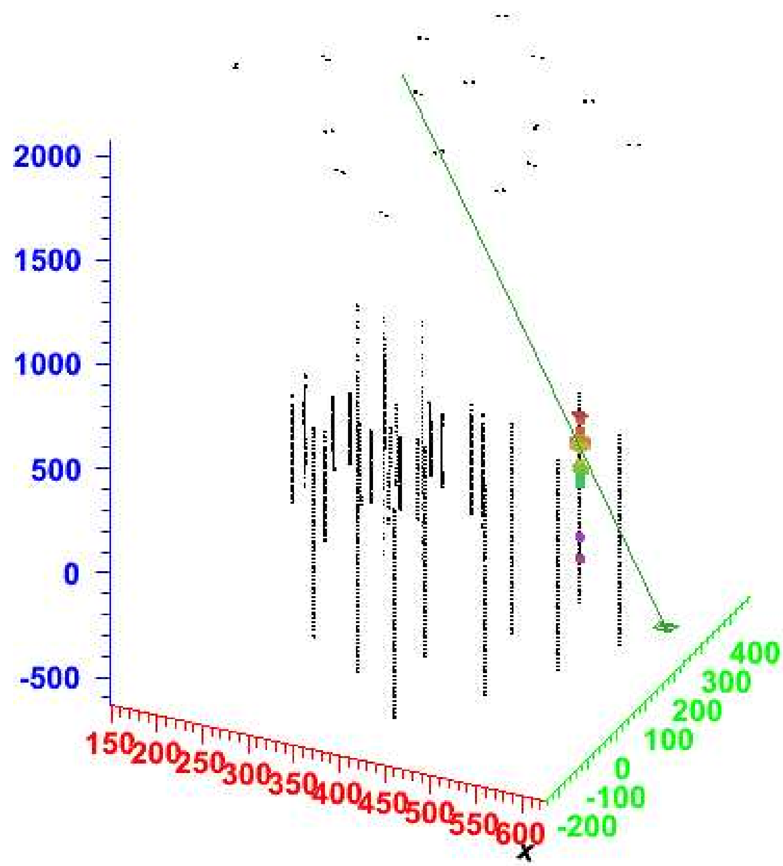

Figure 3.1: *An eventviewer reconstructed muon event for the current configuration of IceCube*

tv__tree- Draw(”fDOM”,””,””, 8836, 0);
tv__tree- Draw(”rStr”,””,””, 19130, 0);

The flasher string number or the reviver string number can be selected, as well as both:

tv__ tree- Draw(”rDOM”,”fStr==30”);
tv__tree- Draw(”rDOM”,”1*(fStr==38 && rStr==38)”,”same”);

The variable ’same’, plots the currently generated graph into the same diagram as before. All following figures were obtained in this manner, in order to extract the detailed information out of the automatically generated output plots. The plots **fStr**, **fDOM** and **rStr** give information about the applied string-DOM set up in the flasher run. The digram **fStr** lists all flashed strings, whereas **rStr** lists the receiver strings. The series of **rDOM** plots display the amount of detected light on the selected receiver string , i.e. **rStr==21**, depending on the DOM number. The colored **rDOM** plot combines all single **rDOM** plots, illustrating the different string contributions to the total amount of detected light. These plots differ in Figure 3.4. compared to the others, because in this flasher run, the receiver strings were also the flasher strings. As a result, the intra-string detection is indicated as blue and the external detection as red. The color plots **dist** and **time** indicate for each receiver DOM, depending on its string, the distance, or time, between flashing and detection. The last plots in each figure put the time and distance plots in relation to each other. These plots represent the relation of a well defined combination of receiver and flasher strings, indicated for example by, (**time:distfStr==30 && rStr==38**).

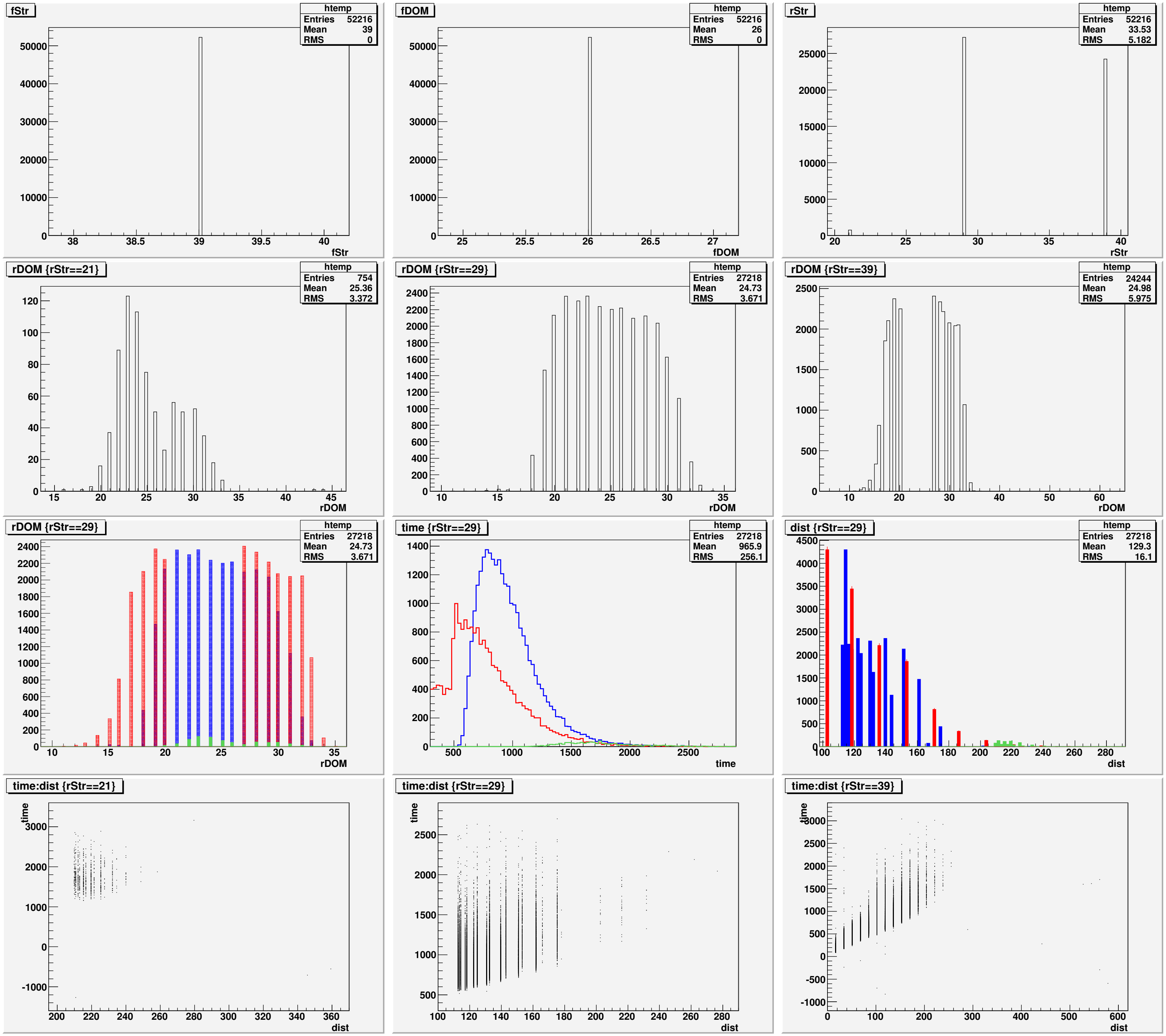

Figure 3.2: *Flasher-verif plots produced with data set 1*

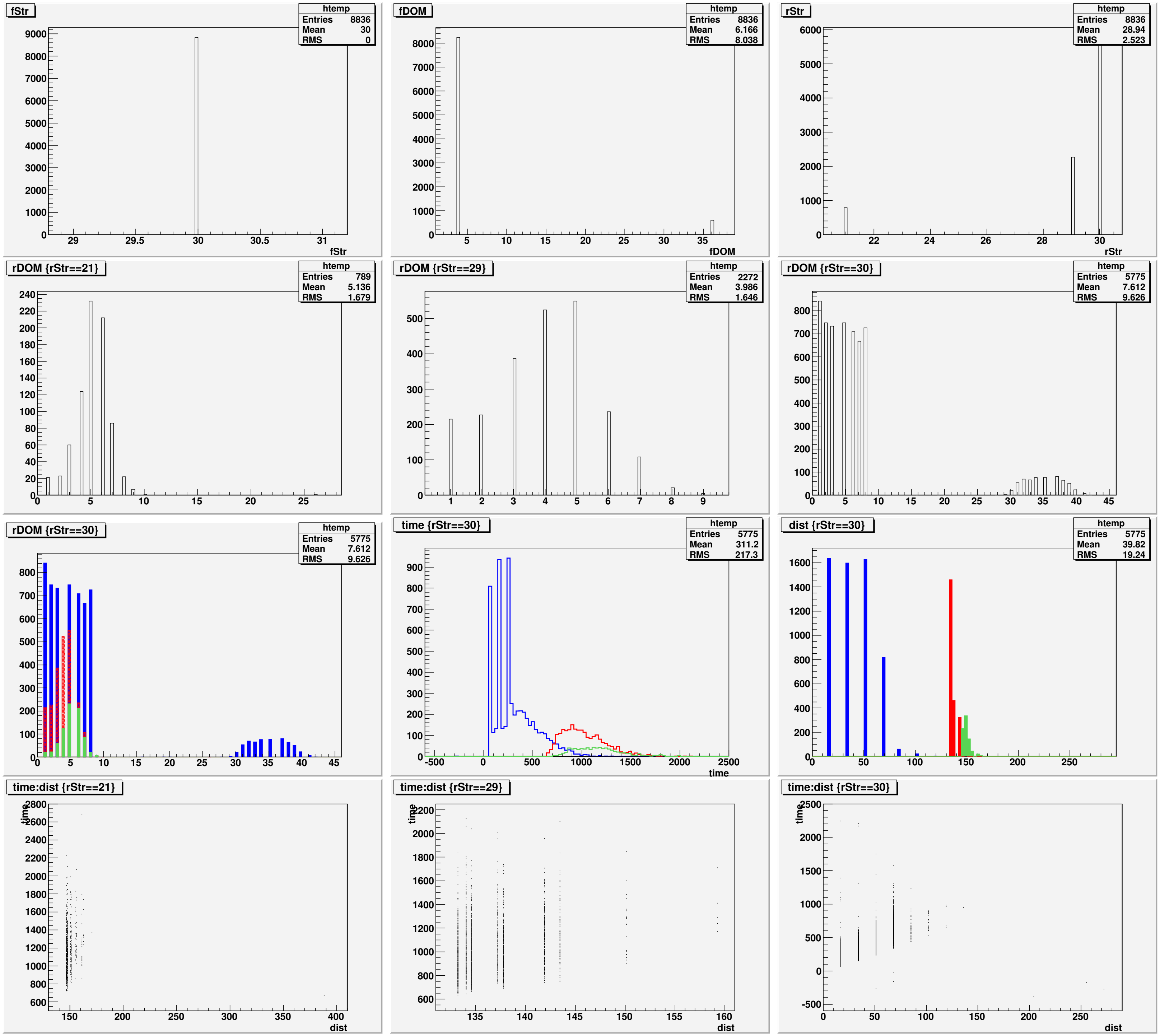

Figure 3.3: *Flasher-verif plots produced with data set 2*

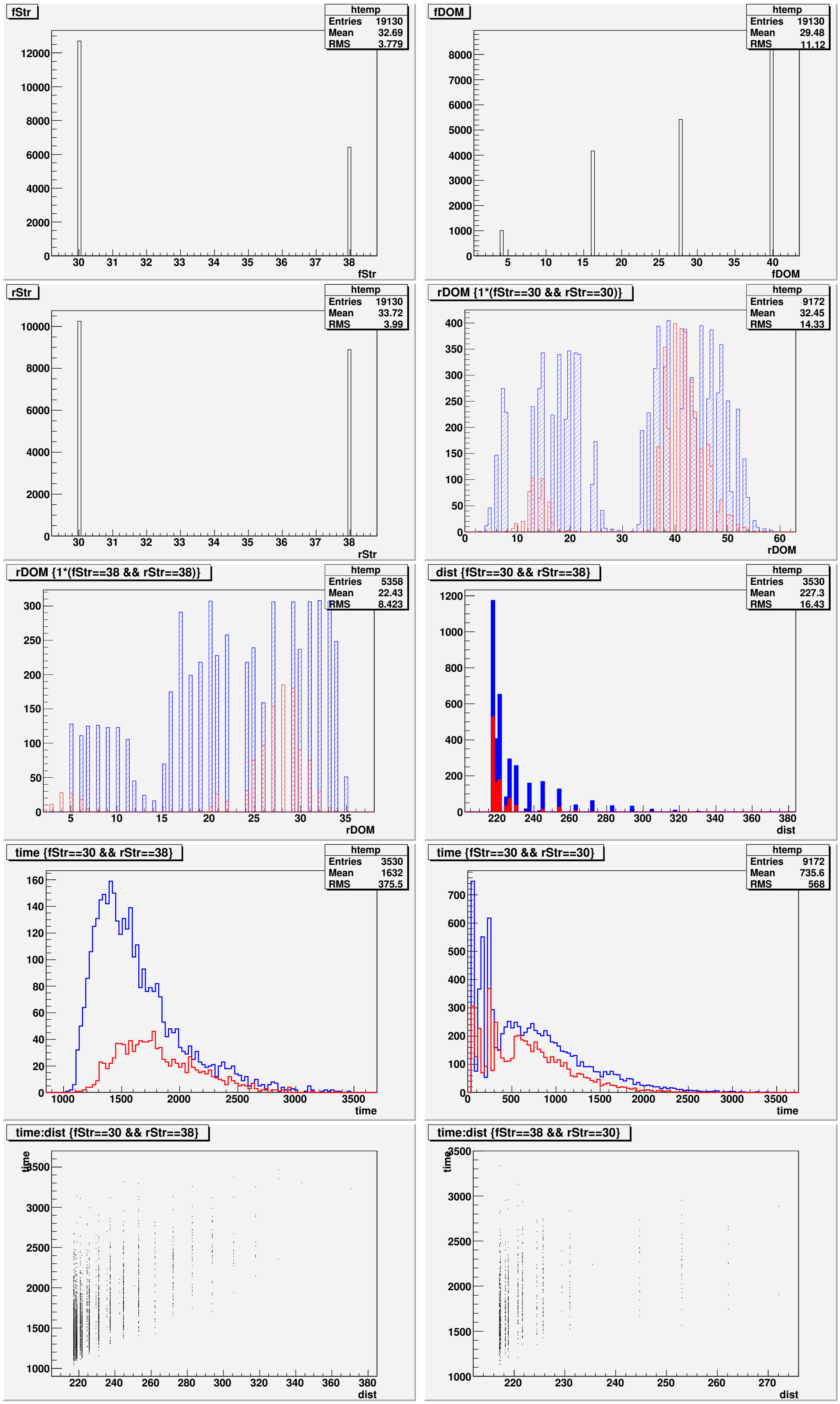

Figure 3.4: *Flasher-verif plots produced with data set 3*

Chapter 4

Interpretation

4.1 Relative String Positions of IceCube 2006 Configuration

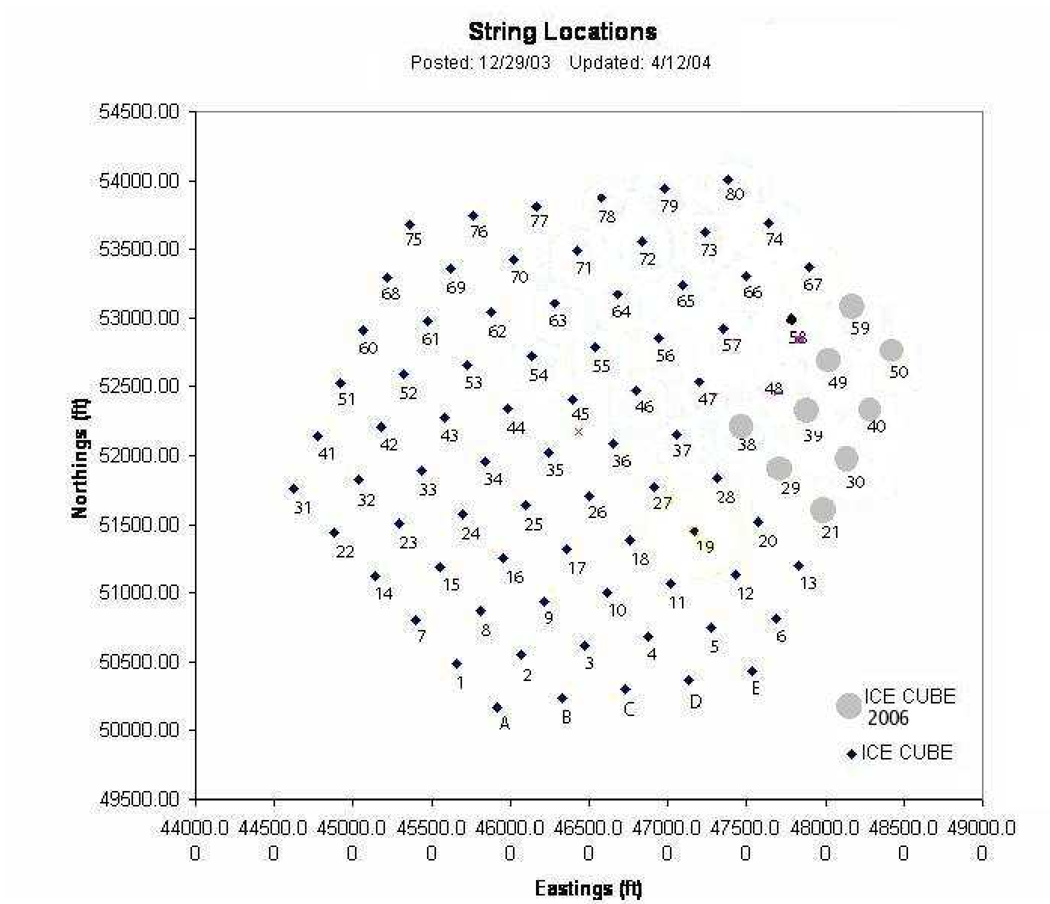

Figure 4.1: *Grid-plan of the complete planned IceCube site; 2006 configuration is marked separately*

The site map of the complete IceCube experiment, including the highlighted 2006 configuration, is shown in Figure 4.1. The three evaluated flasher runs are used to determine the relative distances between the strings. Therefore, the plots, (time) and (distance), of the table of plots in section 3.2, are analyzed. In summary, Figure 4.2 drafts the five strings, which were used in the flasher runs. The distances a, b, c, d and e can be estimated and calculated from the three software runs. Whereas dataset 1 and 3 are reflecting the distance obtained around DOM 25-40, dataset 2 yields a good estimation for the distances around the upper most DOMs, especially DOM 4. All values displayed in the Table below, are taken from the plots in section 3.2 and the distances are given in meters.

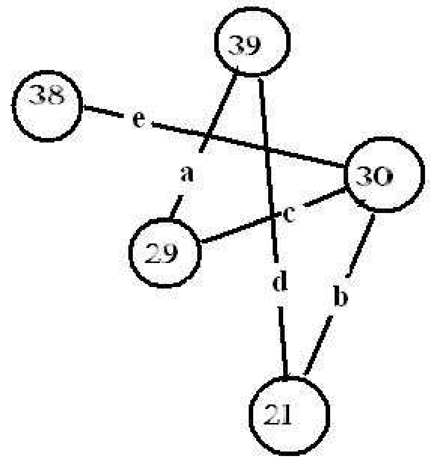

Figure 4.2: *Reconstructed grid from flasher-data*

	Lower Layers		**Upper Layers**	
	Determined Distance	**Actual Distance**	**Determined Distance**	**Actual Distance**
a	121	**125**		
b			145	**125**
c			133	**125**
d	210	**216**		
e	220	**216**		

The optical properties of the ice at the South Pole are strongly depth dependent. Scattering of visible light is due almost entirely to the presence in the ice of insoluble mineral dust grains, sea salt grains, and possibly liquid acid droplets deposited in snow as aerosols and subsequently compressed into the growing ice sheet. Kurt Woschnagg [6] measured the scattering index in depth from 140 m to 2300 m. The upper layers around Dom 4, which is at an average depth of 1520 m, are located according to Kurt Woschnagg's research, in a zone of maximum scattering. Consequently, the string distances obtained in these layers can be explained by the higher probability of scattering. On the other hand the lower layers around 2000 m are located in a zone of minimal scattering. This is underlined by the precision of the values, but lower scattering cannot explain that the two derived distances are shorter than the real distance. Nevertheless, it indicates that the ice properties at greater depth are preferable for Cherenkov Light detection.

4.2 DOM Sensitivity

The detector is set up to detect mainly 'up-coming' neutrinos, crossing through the Earth. To obtain maximum sensitivity, the PMTs are pointing downwards in the direction of the center of the Earth. Thus, the receiving strings should also be more sensitive above the artificially flashed DOMs. By looking at the single **rDOM** plots in section 3.2., one can clearly see a significant drop in the total amount of detected light in the DOMs below the flasher DOM. However, this effect is not observed at each measured distance. Distance b in the upper layer, which was derived to be 145 m, and is therefore the distance with the biggest discrepancy of 20 m, does not show this tendency. This large error in distance can be explained by a higher probability of scattering, which mainly effects the direction of the light. Hence, this unusual trend in Figure 3.3. **rDOM** plot can be interpreted with a lower scattering length in the ice column between string 21 and 30, at around a depth of 1500 m.

Chapter 5

Conclusion

The IceVerV00-02-06 verification software is running and was successfully tested on several test and real runs. The information gained solely from the plots of the flasher-verif project confirm the deployed grid pattern of the strings and approve the different ice zones around 1500 m and 2000 m, extensively researched in [6]. Additionally, more general trends and information, like the DOM sensibility, can be observed in the plots. The **time:dist** plots contain detailed information about refraction indices of the ice depending on its direction. With the determined intra-string data, ice properties such as absorption and scattering, can be researched within the string. Information about the ice column between the strings can be derived from the external string data. Extracting this hidden information from the plots would require changes in the C++ source code of the project, in order to get a more detailed draft of this exact information. Nevertheless, scattering as a fact can be seen in the **time:dist** plots without further changes. The lowest dots in the plots should form a straight line without effects like scattering and absorption. All graphs are describing a decrease in distance gained within the same time for larger distances from the flasher source. Also, the dotted area above the perceived graph line is created due to scattering and increases within the larger values for time and distance.
This project shows that, after a relatively short time, data can be analyzed with significantly high accuracy. The next step in this process is to combine old i3-Modules with a new personal programmed i3-Module and evaluate the researched physics.

Bibliography

[1] Carsten Rott, IceCube: Performance, Status and Future.
arXiv:astro-ph/0611726 v2, 28 November 2006

[2] The IceCube Collaboration, First Year Performance of The IceCube Neutrino Telescope.
arXiv:astro-ph/0604450 v2, 25 April 2006.

[3] Carsten Rott, IceCube: Performance, Status and Future.
arXiv:astro-ph/0611726 v2, 28 November 2006

[4] Simon Patton L.B.N.L., The Frame/Stream/Stop Conceptual Model.
December 25, 2002

[5] IceCube wiki page,
http://wiki.icecube.wisc.edu/index.php/Main$_Page$

[6] Kurt Woschnagg, Optical Properties of South Pole Ice at Depths from 140 to 2300 Meters.
Department of Physics, University of California, Berkeley, CA 94720, USA

Acknowledgments

I would like to thank Dr. Jenni Adams for her supervision in this project. Also, special thanks to Dr. Suruj Seunarine for his help and support with all IT questions.